

## РАЗРАБОТКА СТРУКТУРЫ БАЗЫ ДАННЫХ ДЛЯ ЗАДАЧИ ИНДЕКСАЦИИ ВИДЕОФАЙЛОВ

О.В. Балашова

Научный руководитель: С.В. Аксенов  
Томский политехнический университет  
helga1991@sibmail.com

### Введение

В настоящее время в связи с непрерывным ростом графической информации существует потребность в различных средствах ее обработки. Одной из основных задач в области компьютерного зрения является задача извлечения информации из изображений. В широком смысле анализу подвергаются изображения, начиная с получаемых с помощью самолетных и спутниковых устройств и заканчивая биомедицинскими исследованиями. Применительно к современному пользователю анализу могут подвергаться различные видео файлы, фотографии, содержимое вебсайтов.

Различные средства просмотра файлов (редакторы, файловые менеджеры) позволяют пользователям производить множество манипуляций с видео и фотографиями, в частности хранить их по категориям и осуществлять навигацию. Но что, если пользователь хранит в единой папке огромное количество заранее никак не идентифицированных файлов? В таком случае манипуляция с ними значительно усложняется, и перед разработчиком стоит задача создания средства, упрощающего работу с видео-файлами и изображениями.

### Постановка задачи

Требуется разработать структуру базы данных для некоторой системы. Система представляет своего рода хранилище данных (фотографий или видео), которое на основе содержимого файла присваивает ему идентифицирующий тег. Работу приложения можно продемонстрировать на следующем примере: пусть пользователь хранит в одной папке фотографии с похода в горы и фотографии отдыха на море. Система предусматривает анализ каждого изображения, в ходе которого каждому из них можно присвоить теги, отражающие содержимое файла (для нашего примера это могут быть такие теги как «горы», и «пляж»). Аналогично осуществляется анализ видео по предварительно отобраннным кадрам. В основе функционирования системы лежит текстурный анализ изображения, в частности получение текстурных характеристик, однозначно определяющих принадлежность региона изображения заданной текстуре. В рамках данной работы осуществляется проектирование базы данных для поставленной задачи. Для решения задачи используется концепция реляционных баз данных.

### Реляционная база данных

Реляционная база данных представляет собой множество взаимосвязанных таблиц, каждая из которых содержит информацию об объектах определенного вида. Каждая строка таблицы содержит данные об одном объекте (например, автомобиле, компьютере, клиенте), а столбцы таблицы содержат различные характеристики этих объектов - атрибуты (например, номер двигателя, марка процессора, телефоны фирм или клиентов).

Строки таблицы называются записями. Все записи таблицы имеют одинаковую структуру - они состоят из полей (элементов данных), в которых хранятся атрибуты объекта. Каждое поле записи содержит одну характеристику объекта и представляет собой заданный тип данных (например, текстовая строка, число, дата). Для идентификации записей используется первичный ключ. Первичным ключом называется набор полей таблицы, комбинация значений которых однозначно определяет каждую запись в таблице. [1]

### Выявление сущностей

В предполагаемой базе данных можно выделить 6 сущностей (видеофайл, сцена, кадр, текстура, текстурная характеристика, регион) и 4 кодификатора (формат видео, формат кадра, тип текстуры). Кодификаторы в данном случае служат для упрощения представления (рис. 1). Отношения между выявленными сущностями представлены на рис. (1)

Входное видео состоит из множества сцен, каждая из которых в свою очередь представлена совокупностью кадров. Кадр – это изображение в некотором формате, имеющее различные регионы. Принадлежность региона конкретной текстуре определяется значением его текстурной характеристики.

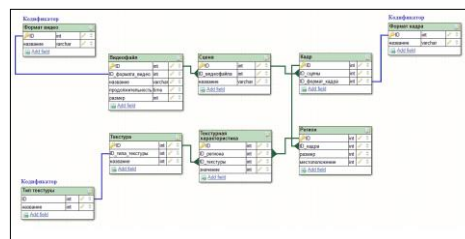


Рис. 4. Диаграмма базы данных

### Описание сущностей и кодификаторов

#### Сущности:

- «Видео» - представляет исходный видеофайл определенного формата. Ключевой атрибут - ID, внешний ключ - ID\_формата\_видео. Каждый видеофайл представлен совокупность сцен, соответственно связь между сущностями «видео» и «сцена» 1:M. Для сущности «видео» используется кодификатор, хранящий названия различных форматов изображения.

- «Сцена» - сцена из видео (совокупность кадров); Ключевой атрибут - ID, внешний ключ - ID\_видеофайла. Каждый сцена представлена совокупность кадров, соответственно связь между сущностями «сцена» и «кадр» 1:M

- «Кадр» - единичное изображение определенного формата; Ключевой атрибут - ID, внешние ключи - ID\_сцены, ID\_формата\_кадра. Для сущности «кадр» используется кодификатор, хранящий названия различных форматов изображения. Кадр может быть разбит на множество регионов, таким образом связь между сущностями «кадр» и «регион» = 1:M

- «Регион» - конкретная область (объект) на кадре; Ключевой атрибут - ID, внешний ключ - ID\_кадра. Регион - это область на кадре, которая в заданной задаче подвергается текстурному анализу. В каждом регионе может быть выделено множество текстурных характеристик. Связь между сущностями «регион» и «текстурная характеристика» = 1:M

- «Текстурная характеристика» - сущность, в которой для определенного региона заданная характеристика имеет определенное значение. Например, в некотором регионе характеристика «линия» имеет значения 100-200. Также данная таблица позволяет определить, какой текстуре принадлежит данный регион. Ключевой атрибут - ID, внешние ключи - ID\_региона, ID\_текстуры.

- «Текстура» - хранит названия текстур. На основе принадлежности региона заданной текстуре и названия текстур формируется название тегов в системе,

#### Кодификаторы:

- «Формат видео» - хранит названия различных форматов видеофайла

- «Формат кадра» - хранит названия форматов цифрового изображения

- «Тип текстуры» - хранит название класса для заданной конкретной текстуры. Например: такие текстуры как «леопард», «кошка» имеют тип «кошачьи», текстуры «волна», «брызги» - тип «вода».

#### **Заключение**

В данной работе была предложена модель базы данных для задачи индексации видеофайлов. Предложенная модель может найти свое применение при необходимости организовать процессы доступа к хранилищу данных. Основным недостатком предложенной реляционной модели является то, что если рассматриваемая задача требует хранения огромного объема данных, то для нее более актуальными являются сверхбольшие базы данных и хранилища данных, которые требуют особых подходов к логическому и системно-техническому проектированию.

#### **Литература**

- 1) Проектирование баз данных. Реляционная база данных и ее структура. [Электронный ресурс], – режим доступа: [http://sernam.ru/book\\_cbd.php?id=2](http://sernam.ru/book_cbd.php?id=2)
- 2) Основы проектирования реляционных баз данных [Электронный ресурс], – режим доступа: <http://citforum.ru/database/dbguide/>
- 3) MySQL Workbench: практическое руководство по проектированию БД. [Электронный ресурс], – режим доступа: <http://codeinlife.ru/web/mysql-workbench-prakticheskoe-rukovodstvo-po-proektirovaniyu-bd.html>